Laboratory 2
MIPS Software Development

Due Date: 25 February 2020 Name:______________________________

Points: 100 Points
Work individually.

Objective: The purpose of this laboratory is to expand your MIPS programing skills by learning to implement nested routines. You will be using the stack to pass operands to, and return results from, a routine. You will also use the stack to save return address information, so that nested routines can be implemented properly. You will generate MIPS assembly language source code that will be assembled, downloaded, and executed. The MIPS instruction set reference sheet that was distributed in class will be very useful for this software development process. (All class slides and reference materials are posted on UNM Learn.)

Activities: For this assignment, you will write a **main** routine in which two n x 1 vectors are defined in a similar manner to how numbers_to_use: is implemented in lab 0***.*** ***(n must be equal to 6 and some of the values must be negative numbers – no values should be 0).*** Your main program will configure the stack pointer appropriately and pass the parameters to a function called **dot_product**. ***Do NOT use registers to pass the parameters to the dot product function and do NOT pass pointers to the arrays. References to the input arguments and output values will all be relative to the stack pointer. You will use the stack to return the dot product result to main, as well as the two vector average values.***

You will generate a function named **dot_product** which takes elements from each vector and calculates the dot product. (Good programming practice will include overflow checks for the multiplication, if you choose to use large value for the vector elements.) The dot_product function will call a function named **average,** which calculates the average value of the elements in each individual vector. Your dot-product routine will need to preserve the return address register on the stack before calling the average function and restore the return address after the average function executes.

You may use the $vx registers to return the value(s) from the function average value.

Upon completion of the dot-product calculations, your main program should have a “spin” loop at the end so that you can use the debugger to verify that your program has correctly calculated the vector dot product.

For this lab, you will be graded on how well you document your code. You will be well served to spend a little time generating pseudocode that shows the high level detail of what your code is supposed to be doing. Spend some time generating the design before you actually start writing code. Include your pseudocode in your lab report.

Your code will be examined to verify that you strictly adhere to the MIPS register usage conventions. **It is mandatory that you use the stack to pass parameters to the called routines as specified above, as this is the one of the main objectives of this lab.**

Documentation: Your lab activities must be documented following the guidelines that are provided on the course UNM Learn site. You must also demonstrate that your project functions properly to one of our TAs, who will then sign your copy of this assignment sheet.

Reference Information: A vector dot product of two vectors, $\bar{X}\ and\ \bar{Y}$, each of which have *n* elements, is defined as follows:

$$\bar{X}\ \cdot \bar{Y} = \ a_n\, b_n + \ a_{n-1}\, b_{n-1} + \cdots .. + a_2\, b_2 + a_1\, b_1$$

Suggestion: Keep all of your files on a USB memory device as there is no guarantee that any information you store on lab machines will be preserved. On occasion, the machines must cleaned and reloaded, so any information stored on them will be lost.